

WHAT IS CLAIMED IS:

3 1. For use in a redundant, high-availability system of
4 processor-based components, a system for memory equalization
5 comprising:

6 a memory containing data elements each stored within one
7 of a plurality of defined memory segments, wherein the data
8 elements within the memory segments may be selectively changed;

9 a direct memory access circuit capable of automatically
10 copying memory segments from the memory to a queue;

11 a data link coupled to the queue, wherein each of the
12 plurality of memory segments is structured to form a data packet
13 which may be transmitted without internal changes over the data
14 link; and

15 a control preventing the memory segments from receiving
16 changes to the data elements contained therein at a rate faster
17 than a transfer rate of memory segments over the data link.

1 2. The system as set forth in Claim 1, further comprising:
2 a processor intermittently changing data elements within
3 the memory segments, wherein the data link operates without direct
4 control by the processor.

1 3. The system as set forth in Claim 1, wherein the memory
2 further comprises a set of memory segments mapping to a
3 corresponding set of memory segments within a device coupled to the
4 data link.

1 4. The system as set forth in Claim 1, wherein the memory
2 further comprises a portion of Firewire Global Memory and wherein
3 the data link comprises a Firewire data link.

1 5. The system as set forth in Claim 1, wherein each of the
2 data elements further comprise one of the following:
3 call state information for a call being processed;
4 resource allocation records for resources allocated to a
5 call being processed; and
6 other information regarding a call being processed.

1 6. The system as set forth in Claim 1, wherein the direct
2 memory access circuit copies memory segments containing changed
3 data elements therein from the memory to the queue, the copying of
4 a memory segment being triggered by a processor which inter-
5 mittently changes data elements within the memory segments.

1 7. The system as set forth in Claim 6, wherein the processor
2 writes a direct memory access descriptor for a memory segment
3 containing a changed data element and sets controls bits initiating
4 copying of the memory segment by the direct memory access circuit.

1 8. The system as set forth in Claim 1, wherein the direct
2 memory access circuit copies memory segments containing changed
3 data elements therein from the memory to the queue, the copying of
4 a memory segment being triggered by a circuit monitoring writes to
5 preselected memory addresses to detect changes to a data element
6 within any memory segment.

1 9. The system as set forth in Claim 1, wherein the direct
2 memory access circuit sequentially copies each of the memory
3 segments from the memory to the queue in a continuous loop.

1 10. The system as set forth in Claim 1, wherein the data link
2 comprises an asynchronous transfer mode switch.

1 11. A redundant, high-availability system of processor-based
2 components, comprising:

3 an active component;

4 a standby component; and

5 a system for memory equalization between the active and
6 standby components comprising:

7 counterpart memories within the active and standby
8 components each containing data elements stored within one of
9 a plurality of defined memory segments mapped to addresses
10 within both of the counterpart memories, wherein the data
11 elements within the memory segments may be selectively
12 changed;

13 a direct memory access circuit within each of the
14 active and standby components, the direct memory access
15 circuit within the active component capable of automatically
16 copying memory segments from the memory within the active
17 component to a queue within the active component and the
18 direct memory access circuit within the standby component
19 capable of automatically copying memory segments from a queue
20 within the standby component to the memory within the standby
21 component;

22 a data link coupling to the queue within the active

23 component to the queue within the standby component, wherein
24 each of the plurality of memory segments is structured to form
25 a data packet which may be transmitted without internal
26 changes over the data link; and

27 a control within the active component preventing the
28 memory segments within the memory in the active component from
29 receiving changes to the data elements contained therein at a
30 rate faster than a transfer rate of memory segments over the
31 data link.

12. The system as set forth in Claim 11, wherein the active
component further comprises:

a processor intermittently changing data elements within
the memory segments, wherein the data link operates without direct
control by the processor.

13. The system as set forth in Claim 11, wherein the
counterpart memories within the active and standby components each
further comprise a set of memory segments mapping to a common set
of addresses.

1 14. The system as set forth in Claim 11, wherein the
2 counterpart memories within the active and standby components
3 further comprise a portion of Firewire Global Memory and wherein
4 the data link comprises a Firewire data link.

1 15. The system as set forth in Claim 11, wherein each of the
2 data elements within either of the counterpart memories further
3 comprise one of the following:

4 call state information for a call being processed;

5 resource allocation records for resources allocated to a
6 call being processed; and

7 other information regarding a call being processed.

1 16. The system as set forth in Claim 11, wherein the direct
2 memory access circuit within the active component copies memory
3 segments containing changed data elements therein from the memory
4 within the active component to the queue within the active
5 component, the copying of a memory segment being triggered by a
6 processor within the active component which intermittently changes
7 data elements within the memory segments within the active
8 component.

1 17. The system as set forth in Claim 16, wherein the
2 processor within the active component writes a direct memory access
3 descriptor for a memory segment within the active component which
4 contains a changed data element and sets controls bits initiating
5 copying of the memory segment within the active component by the
6 direct memory access circuit within the active component.

1 18. The system as set forth in Claim 11, wherein the direct
2 memory access circuit within the active component copies memory
3 segments containing changed data elements therein from the memory
4 within the active component to the queue within the active
5 component, the copying of a memory segment being triggered by a
6 circuit within the active component which monitor writes to
7 preselected memory addresses to detect changes to a data element
8 within any memory segment within the active component.

1 19. The system as set forth in Claim 11, wherein the direct
2 memory access circuit within the active component sequentially
3 copies each of the memory segments from the memory within the
4 active component to the queue within the active component in a
5 continuous loop.

1 20. The system as set forth in Claim 11, wherein a controller
2 for the data link within the standby component verifies data
3 integrity for data packets received over the data link and
4 acknowledges successful transfer of data packets over the data link
5 to the active component.

1 21. The system as set forth in Claim 11, wherein the direct
2 memory access circuit within the standby component automatically
3 moves memory segments from a queue within the standby component
4 coupled to the data link into a corresponding memory location in
5 the memory within the standby component.

1 22. The system as set forth in Claim 11, wherein the data
2 link comprises an asynchronous transfer mode switch.

1 23. The system as set forth in Claim 11, wherein the active
2 component is one of N active components supported by the standby
3 component.

1 24. For use in a redundant, high-availability system of
2 processor-based components, a method of memory equalization
3 comprising the steps of:

4 selectively changing data elements each stored within one
5 of a plurality of defined memory segments contained within a
6 memory;

7 automatically copying memory segments from the memory to
8 a queue utilizing a direct memory access circuit;

9 transferring memory segments over a data link coupled to
10 the queue, wherein each of the plurality of memory segments is
11 structured to form a data packet which may be transmitted without
12 internal changes over the data link; and

13 inhibiting the memory segments from receiving changes to
14 the data elements contained therein at a rate faster than a
15 transfer rate of memory segments over the data link.

1 25. The method as set forth in Claim 24, further comprising:

2 intermittently changing data elements within the memory
3 segments utilizing a processor, wherein the data link operates
4 without direct control by the processor.

1 26. The method as set forth in Claim 24, further comprising:
2 mapping the memory segments to a common set of addresses
3 for counterpart memory segments within a standby component.

1 27. The method as set forth in Claim 26, wherein the steps of
2 mapping the memory segments to a common set of addresses for
3 counterpart memory segments within a standby component and
4 transferring memory segments over a data link coupled to the queue
5 further comprise:

6 mapping a portion of Firewire Global Memory to the memory
7 segments and the counterpart memory segments; and
8 transferring memory segments over a Firewire data link.

1 28. The method as set forth in Claim 24, further comprising:
2 storing one of the following within each of the data
3 elements:

4 call state information for a call being processed;
5 resource allocation records for resources allocated
6 to a call being processed; and
7 other information regarding a call being processed.

1 29. The method as set forth in Claim 24, wherein the step of
2 automatically copying memory segments from the memory to a queue
3 utilizing a direct memory access circuit further comprises:

4 initiating copying of a memory segment by a processor
5 within an active component containing the memory segment, wherein
6 the processor intermittently changes data elements within the
7 memory segments within the active component.

1 30. The method as set forth in Claim 29, further comprising:

2 writing a direct memory access descriptor for a memory
3 segment within the active component which contains a changed data
4 element; and

5 setting controls bits initiating copying of the memory
6 segment which contains the changed data element by the direct
7 memory access circuit.

1 31. The method as set forth in Claim 24, wherein the step of
2 automatically copying memory segments from the memory to a queue
3 utilizing a direct memory access circuit further comprises:

4 initiating copying of a memory segment by a circuit
5 within an active component containing the memory segment, wherein
6 the circuit monitor writes to preselected memory addresses to
7 detect changes to a data element within any memory segment within
8 the active component.

1 32. The method as set forth in Claim 24, wherein the step of
2 automatically copying memory segments from the memory to a queue
3 utilizing a direct memory access circuit further comprises:

4 utilizing the direct memory access circuit, sequentially
5 copying each memory segment from the memory to the queue in a
6 continuous loop.

1 33. The method as set forth in Claim 24, wherein the step of
2 transferring memory segments over a data link coupled to the queue
3 further comprises:

4 transmitting the memory segments to a standby component
5 coupled to the data link, wherein a controller for the data link
6 within the standby component verifies data integrity for data
7 packets received over the data link and acknowledges successful
8 transfer of data packets over the data link to an active component
9 containing the queue.

10 34. The method as set forth in Claim 33, wherein the step of
11 transmitting the memory segments to a standby component coupled to
12 the data link further comprises:

13 receiving the memory segments in a queue within the
14 standby component, wherein a direct memory access circuit within
15 the standby component automatically moves memory segments from the
16 queue within the standby component into a corresponding memory
17 location in a memory within the standby component.

1 35. The method as set forth in Claim 24, wherein the step of
2 transferring memory segments over a data link coupled to the queue
3 further comprises:

4 transmitting the memory segments utilizing an
5 asynchronous transfer mode switch.

1 36. The method as set forth in Claim 24, further comprising:
2 supporting processing within an active component
3 containing the memory and the queue utilizing a standby component
4 coupled to the data link, wherein the active component is one of N
5 active components supported by the standby component.